Hop Reservation Multiple Access (HRMA) for Multichannel Packet Radio Networks*

Zhenyu Tang and J. J. Garcia-Luna-Aceves
Computer Engineering Department
School of Engineering
University of California
Santa Cruz, CA 95064, USA
ktang, jj@cse.ucsc.edu

Abstract

A new multichannel MAC protocol called Hop Reservation Multiple Access (HRMA) for packet-radio networks is introduced, specified and analyzed. HRMA is based on very-slow frequency-hopping spread spectrum (FHSS) and takes advantage of the time slotting necessary for frequency hopping. HRMA allows a pair of communicating nodes to reserve a frequency hop (channel) using a hop reservation and handshake mechanism on every hop to guarantee collision-free data transmission in the presence of hidden terminals. HRMA provides a baseline to offer QoS in ad-hoc networks based on simple half-duplex slow FHSS radios. We analyze the throughput achieved in HRMA for the case of a fully-connected network assuming variable-length packets, and compare it against an ideal multichannel access protocol and the multichannel slotted ALOHA protocol. The numerical results show that HRMA can achieve much higher throughput than multichannel slotted ALOHA in the traffic-load ranges of interest, especially when the average packet length is large compared to a slot size, in which case the maximum throughput of HRMA is close to what can be obtained with an ideal protocol.

1. Introduction

Because of the recent affordability of commercial radios and controllers based on microprocessors, multi-hop packet radio networks (i.e., ad-hoc networks) are likely to play an important role in computer communications. Ad-hoc networks extend packet switching technology into environments with mobile users, can be installed quickly in emergency situations, and are self configurable, which makes them very attractive in many applications, including the seamless extension of the Internet to the wireless, mobile environment.

The unlicensed nature of ISM bands makes them extremely attractive for ad hoc networks; furthermore, there is widespread availability of commercial, affordable radios for the 915MHz, 2.4GHz and 5.8GHz bands. Accordingly, developing medium access control (MAC) protocols with which the nodes (packet-radios) of ad-hoc networks can share the ISM bands efficiently is critical for the future success of such networks.

In ISM bands, radios must operate using direct-sequence spread spectrum (DSSS) or frequency-hopping spread spectrum (FHSS) [2]. This paper focuses on the design of an efficient MAC protocol for ad-hoc networks based on FHSS radios operating in ISM bands.

The maximum dwell time allowed in ISM bands is 400 msec [2], which at 1Mbps allows entire packets to be transmitted within the same frequency hop. On the other hand, keeping the sender and receiver synchronized on the same frequency hops while a packet is being transmitted is not simple when nodes move and data rates are high (1Mbps). Commercially-available radios for ISM bands today are able to synchronize on a packet by packet basis, but not on bit by bit basis. Given the FCC regulations for ISM bands and the characteristics of today's COTS radios, the problem of designing MAC protocols that use very slow frequency hopping (i.e., an entire packet is sent in the same hop) as a combination of time and frequency division multiplexing of the radio channel is very timely. Curiously, there is little work reported on this subject.

There are many prior examples of MAC protocols for frequency-hopping radios, which are typically based on applying ALOHA or slotted ALOHA using the same hopping sequence for all nodes or sender- or receiver-oriented code assignments [6, 8]. However, these approaches assume that radios hop frequencies within the same packet frequently to achieve code division multiple access (CDMA). IEEE 802.11 [1] incorporates a convergence layer that makes the characteristics of the physical layer transparent to the MAC protocol. A concrete example of using very-slow frequency-hopping radios is the MAC protocol used in Metricom's Ricochet wireless data network [3], which assumes that each receiver has its own frequency hopping sequence and makes the sender learn the hopping sequence of the receiver. The sender synchronizes with the receiver's hopping sequence and transmits all its data packet over the same frequency hop on which the receiver is tuned. The data packet can last longer than a frequency-hop dwell time. However, neither [1] nor [3] is exempt from hidden-d terminal interference.

We introduce the Hop Reservation Multiple Access (HRMA) protocol, which takes advantage of the time-slotting properties of slow FHSS. Section 2 specifies HRMA in detail. HRMA uses a common hopping sequence and permits a pair of nodes to reserve frequency hops over which they can communicate without interference. A frequency hop is reserved by contention through a request-to-send/clear-to-send exchange between sender and receiver. A successful exchange leads to a reservation, and each reserved hop starts with a reservation packet from sender and receiver that prevents other nodes from attempting to use the hop. A common frequency hop is used to permit nodes to synchronize with one another, i.e., agree on the current hop of the sequence and the beginning time of a frequency hop. After a hop is reserved, a sender is able to transmit data beyond the dwell time of the reserved hop. In this section, we also demonstrate that HRMA guarantees that no data or acknowledgment packets from a source and receiver collide with any other packets in the presence of hidden terminals.

Section 3 analyzes the throughput of HRMA, an ideal protocol in which it is assumed that one of multiple senders competing for a receiver is always allowed to capture the receiver, and the multichannel slotted ALOHA protocol with receiver-oriented channel assignment (ROCA), for the case of a fully-connected network and variable-length packets. Although our analysis focuses on fully-connected networks for simplicity, it is relevant for comparative purposes, because HRMA does not suffer from hidden-terminal interference, which makes the fully-connected network a worst-case scenario for HRMA from the standpoint of channel reuse.

Section 4 presents the numerical results of our analysis comparing the three protocols; the results show that HRMA achieves very high throughput for the range of traffic load within which the network is stable, which can be enforced in practice with simple backoff strategies. Section 5 presents our conclusions.

*This work was supported by the Defense Advanced Research Projects Agency (DARPA) under grant F30602-97-2-0338.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **1998**	2. REPORT TYPE	3. DATES COVERED **00-00-1998 to 00-00-1998**

4. TITLE AND SUBTITLE **Hop Reservation Multiple Access (HRMA) for Multichannel Packet Radio Networks**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of California at Santa Cruz,Department of Computer Engineering,Santa Cruz,CA,95064**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**8**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

```
Variable Definitions
    SL = Slot Length
    FL = Frame Length
    PD = Packet Detected
    TE = Timer Expired
    LD = Local Data
    Tprop = Maximum Propagation Delay
    Tproc = Processing Delay

Procedure START()
Begin
    Timer ← 3 × FL;
    While ($\overline{PD}$ ∧ $\overline{TE}$) listen;
    If(PD)
    Then Begin
        Receive packet;
        DO CACE of (received packet type)
        Begin
            Synchronizing packet:
                Accept the synchronizing parameters;
                Call PASSIVE();
            Default:
                Call START();
        End
    End
    Else Begin
        Set the synchronizing parameters and send synchronizing packet;
        Call PASSIVE();
    End
End

Procedure ACCESS ()
Begin
    Send RTS to destination in RTS period;
    Listen during CTS period;
    If (CTS received for the RTS sent) Then call XMIT ();
    Else call BACKOFF ();
End

Procedure PASSIVE()
Begin
    DO CASE of (event type)
    Begin
        Beginning of Synch period:
            Call SYNCH ();
        During HR period:
            Listen;
        End of HR period:
            If (LD in Synch or HR period ∧ $\overline{PD}$ in HR period)
            Then call ACCESS ();
        During RTS period:
            Listen;
        Beginning of CTS period:
            If (RTS received ∧ Destination ID = Local ID)
            Then Begin
                send CTS to source;
                Call RECEIVE ();
            End
        End of CTS period:
            If ((LD in RTS or CTS period) ∨ (LD ∧ PD in HR period))
            Then call BACKOFF ();
            Else call PASSIVE ();
    End
End

Procedure RECEIVE ()
Begin
    More_Packet ← 1;
    HR_Timer ← FL;
    While (HR_Timer not expired ∧ More_Packet)
    Begin
        Timer ← 2 × Tprop + Tproc;
        While ($\overline{TE}$ ∧ $\overline{PD}$) wait;
        If (TE)
        Then Begin
            If (LD) Then call BACKOFF ();
            Else call PASSIVE ();
        End
        Else Begin
            Receive Packet;
            DO CASE of (received packet type)
            Begin
                DATA:
                    If (Destination ID = Local ID)
                    Then Begin
                        Pass packet to upper layer;
                        If (last packet)
                        Then Begin
                            More_Packet ← 0;
                            Hop to the corresponding ack frequency;
                            Send Ack to source;
                        End
                    End
                    Else Begin
                        If (LD) Then call BACKOFF ();
                        Else call PASSIVE ();
                    End
                Default:
                    If (LD) Then call BACKOFF ();
                    Else call PASSIVE ();
            End
        End
    End
    If ($\overline{More\_Packet}$)
    Then Begin
        If (LD) Then call BACKOFF ();
        Else call PASSIVE ();
    End
    Else Begin
        Send HR packet in the next HR period;
        Listen during RTS and CTS period;
        Call RECEIVE () at the end of CTS period;
    End
End

Procedure SYNCH()
Begin
    Hop to f0;
    While (in Synch period) do synch information exchange;
    Hop to the next frequency according the hopping pattern;
    Return to calling procedure;
End

Procedure XMIT ()
Begin
    If (data length > FL)
    Then Begin
        Construct a data packet of length FL by taking part of the data;
        More_Packet field in packet header ← 1;
    End
    Else Begin
        Put all the data in a packet;
        More_Packet field in packet header ← 0;
    End
    Transmit data packet;
    Timer ← delay till the end of the next HR period;
    If (No more data) Then hop to the corresponding ack frequency;
    While ($\overline{TE}$ ∧ $\overline{PD}$) Listen;
    If (PD)
    Then Begin
        Receive packet;
        DO CASE of (event type)
        Begin
            More data ∧ HR received for this link:
                Remove the data sent from data buffer;
                Send RTS;
                Call XMIT () at beginning of next slot;
            No more data ∧ Ack received for this packet:
                Remove the data sent from data buffer;
                If (LD) call BACKOFF();
                Else call PASSIVE();
            Default:
                Call BACKOFF();
        End
    End
    Else call BACKOFF ();
End

Procedure BACKOFF()
Begin
    Calculate maximum number of backoff slots, m,
    according to some backoff algorithm;
    Timer ← RANDOM(0, m × SL);
    While (TRUE) Do
    Begin
        DO CASE of (event type)
        Begin
            Beginning of Synch period:
                Call SYNCH();
            During HR period:
                Listen;
            End of HR period:
                If (TE ∧ $\overline{PD}$ in HR period) Then call ACCESS ();
            During RTS period:
                Listen;
            Beginning of CTS period:
                If (RTS received ∧ Destination ID = Local ID)
                Then Begin
                    Send CTS to source;
                    Call RECEIVE ();
                End
                Else listen during CTS period;
            End of CTS period:
            If (TE ∧ PD in HR period) Then call BACKOFF ();
        End
    End
End
```

Figure 1: HRMA Specification

2. HRMA Protocol

HRMA is based on a common hopping sequence for the entire network and requires half-duplex slow frequency-hopping radios with no carrier sensing to operate. HRMA can be viewed as a time-slot reservation protocol in which a time slot is also assigned a separate frequency channel.

The pseudo code in Figure 1 presents the specification of HRMA. We note that the mechanism used to contend for and reserve frequency hops in HRMA is similar in complexity to such simple MAC protocols as FAMA [4][7] and MACAW [5]. To simplify our analysis, a non-persistent policy is used for hop reservations; persistent versions of HRMA are also possible.

2.1. Synchronizing Nodes to a Common Hopping Sequence

As in any MAC protocol operating with FHSS radios, time in HRMA is slotted. HRMA uses one of the L available frequencies, which we denote by f_0, as a dedicated synchronization channel on which all nodes exchange synchronization information to agree on the beginning of a frequency hop and the current hop. The rest of the frequencies are organized into $M = \lfloor (L-1)/2 \rfloor$ frequency pairs $(f_i, f_i^*), i = 1, 2, ..., M$. For any node i, the frequency hop f_i is used for sending or receiving hop-reservation (HR) packets, request-to-send (RTS) packets, clear-to-send (CTS) packets, and data packets, while frequency hop f_i^* is used for sending or receiving acknowledgments to data packets sent on frequency hop f_i.

Each HRMA slot consists of one synchronization period, one HR period, one RTS period and one CTS period, each of which is used exclusively to send or receive a synchronization packet, an HR packet, an RTS packet, and a CTS packet, respectively. All the nodes that are not transmitting or receiving data packets must hop to the synchronization frequency f_0 and exchange synchronization messages during the synchronization period of each slot. The frequency on which nodes may tune their radios during the HR, RTS and CTS periods remains the same in each slot and varies among $f_i, i = 1, 2, ..., M$, slot by slot according to the common frequency hopping sequence.

A special slot is defined that is of the same size as a normal slot, but consists of only one synchronization period fixed on f_0 for synchronization purpose. This special synchronization slot followed by M consecutive normal slots, which pass through all the M frequencies in the hopping sequence, makes up a HRMA frame, as shown in Figure 2.

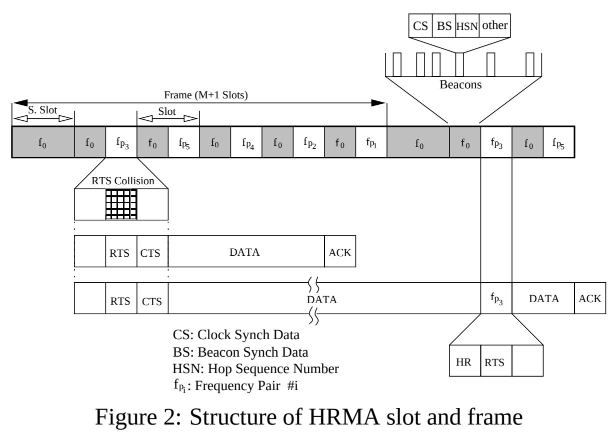

Figure 2: Structure of HRMA slot and frame

When a new node becomes operational, it must listen to the synchronization channel for a time period long enough to gather the synchronization information about hopping pattern and timing of the system, so that it can get synchronized with the system. If the new node does not detect any synchronization information during that time, it finds an empty system. The new node can broadcast its own synchronization information and create a new one-node system. A new node can easily join or create a system with HRMA, because the synchronization information is repeated in every HRMA slot. Hence, nodes in the same connected component of a network, which we call group, are synchronous with each other. In contrast, nodes from different groups are disconnected and asynchronous.

Let the length of a HRMA slot and the synchronization period of a normal HRMA slot be η and η_s, respectively. It can be seen from Figure 3 that the dwell time of f_0 at the beginning of each frame is $\eta + \eta_s$. Because the synchronization period is repeated at the beginning of each HRMA slot, there must be at least one f_0 synchronization period of length η_s within any interval of length $\eta + \eta_s$. Therefore, any two nodes from disconnected groups must always have at least two overlapping time periods of length η_s on f_0 within any time period equal to the duration of a HRMA frame no matter how large the timing offset between the different groups is. Figure 3 shows the worst case overlapping time between asynchronous systems. Therefore, HRMA allows different groups to merge.

A synchronization protocol based on a listen-before-transmit policy for beacon packets similar to that advocated in IEEE 802.11 [1] can be used in the synchronization periods. However, it would be difficult for asynchronous groups to merge in 802.11 networks, because there is no equivalent to our proposed synchronization slot and frequency hop.

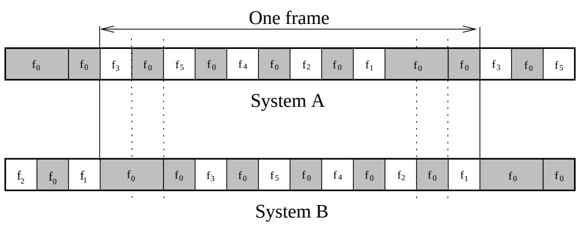

Figure 3: Worst case overlapping time on synchronization frequency

2.2. Accessing and Reserving Hops

Assuming that nodes are able to synchronize according to a common hopping sequence, the rest of HRMA's operation pertains to the way in which nodes access and reserve specific frequency hops. All nodes that are not communicating with other nodes over reserved hops are hopping together, and we assume a non-persistent approach to hop reservation in HRMA.

When an idle node receives a data packet to transmit before the RTS period of a given slot has started, the node backs off if the HR period contains an HR packet. The back-off time is random and is a multiple of the HRMA slot time, so that the node is ready to attempt transmission at the beginning of a slot after the back-off time elapses. Otherwise, if there is no HR packet claiming the slot, the node sends an RTS to the intended receiver and waits for the CTS, which must come from the receiver during the CTS period. Whenever a node receives an RTS intended for it, it sends a CTS back to the source in the CTS period of the same slot and stays in the same frequency channel waiting for any data packet. If a node that sent an RTS receives no CTS from the receiver in the CTS period, it backs off a random number of slots and tries to send its RTS again in another slot. The source node transmits a data packet if it receives a CTS from the receiver; the data packet is transmitted in the same frequency channel used for the RTS-CTS exchange.

When an idle node receives a data packet to transmit after the RTS period of a given slot has started, the node simply backs off. This is done because such a node is unable to request the current slot anymore.

After the CTS period of a slot, the next slot starts and all nodes that are not transmitting or receiving data packets hop to f_0 and dwell on f_0 for a period of length η_s to exchange synchronization information, and then hop to the next frequency hop of the M frequencies in the common hopping pattern.

A data packet transmitted in HRMA can be of any length and the node can send multiple data packets as well. However, because HRMA operates in the ISM band, a data packet or packet train cannot exceed the maximum dwell time allowed by the FCC [2]. The source and destination dwell on the same frequency hop during the entire data packet or packet train.

When the data that need to be exchanged between sender and receiver require multiple HRMA frames for their transmission, the sender notifies the receiver and the receiver sends an HR packet during the HR period of the same slot of the next frame. This informs the neighbors of the receiver that they cannot attempt to send RTSs in the HRMA slot occupied by sender and receiver. When the sender receives the HR from the receiver, it sends an RTS to jam any possible RTSs addressed to its own neighbors, which may not hear the receiver. Thus, without further contention, the frequency hop is reserved again by the sender and receiver for the following HRMA frame. Both sender and receiver are silent in the CTS period of the slot, and more data are transmitted after that period over the same frequency channel of the HRMA slot. The frequency hop remains reserved in a similar fashion, until the sender relinquishes it.

After the source sends a data packet, it transitions to the acknowledgment frequency defined according to the frequency on which the data packet was sent, and the receiver sends an acknowledgment packet back to the source on that acknowledgment frequency.

The different cases for access and reservation of hops are shown in Figure 2.

A more efficient variant of HRMA allows the data including piggybacked acknowledgment to flow in both directions and establishs a duplex data pipe between a pair of nodes, with one node transmitting on f_i and the other on f_i^*. With this approach, the same hop reservation procedure is needed whenever the data in either direction last longer than an HRMA frame.

2.3. Correctness of HRMA

The following theorem proves that HRMA eliminates hidden-terminal interference problems. To prove the theorem, we assume that all nodes are synchronized, that there is no capture effect on any channel, and that any overlap of transmissions at any receiver on any channel causes all packets to be lost. We assume that links are bidirectional, which is a requirement that stems from the RTS/CTS exchange used to reserve frequency hops.

A neighbor of a node A is a node that has a link to A. All the neighbors of node A are denoted by the set $N(A)$.

Theorem: HRMA guarantees that no data or acknowledgement packet collides with any other packet in the presence of hidden terminals.

Proof: If no RTS is successful, then no data packet or acknowledgment packet is sent and thus no data or acknowledgment packet is involved in any collision.

If a destination node D successfully receives an RTS from a source node S on frequency hop f_k in slot m, it must be true that no node other than S in $N(D)$ is transmitting on f_k in the RTS period of slot m; otherwise, there will be a collision of RTSs at the destination D. Therefore, no other node in $N(D)$ can be a source node on f_k during the following HRMA frame. However, note that any other node in $N(D)$ could be or become a successful destination on hop f_k if it is not in $N(S)$. It must also be true that no node other than D in $N(S)$ can receive a correct RTS addressed to it in slot m; for otherwise the RTS from S would interfere with it. Accordingly, no node other than D in $N(S)$ can be or become a successful destination on hop f_k during the following HRMA frame, but it can be or become a successful source on f_k if it is not in $N(D)$. As a result, during the following HRMA frame, S is the only source on f_k in $N(D)$ and D is the only successful destination on f_k in $N(S)$. Therefore, the CTS from D and data packet(s) from S are collision free.

If the data packet lasts longer than a frame, the destination sends an HR in the same (frequency) slot (slot m) of the next frame, which prevents any node in $N(D)$ from sending an RTS on f_k (in slot m) and becoming a source node. HR is collision free at S, because R is the only destination on f_k in $N(S)$. After S receives an HR, it sends an RTS on the same frequency f_k (in slot m), and this prevents any node in $N(S)$ from correctly receiving any possible RTS on f_k directed for that node and becoming a destination. Therefore, it is true that S is the only source on f_k in $N(D)$ and D is the only successful destination on f_k in $N(S)$ during another HRMA frame. Also note that nodes in $N(S)$ but not in $N(D)$ can become successful sources and nodes in $N(D)$ but not in $N(S)$ can become successful destinations on f_k during this following one-frame-long period of time. Therefore, a data packet from S will be collision free in any subsequent HRMA frame, until the end of the data.

The ack packet for a data packet is sent on a different frequency of the corresponding frequency pair, f_k^*; therefore, an ack packet can only collide with other ack packets. However, as stated above, no two successful destinations exist in the neighborhood of any successful source on the same frequency hop, which implies no ack packet can collide with any other ack packet.

It follows from the above that HRMA guarantees that data and ack packets are free of collision in the presence of hidden terminals. □

3. Comparative Throughput Analysis

3.1. System Model and Assumptions

For simplicity, we assume a fully-connected network, which corresponds to a worst-case scenario from the standpoint of channel reuse in HRMA, because HRMA ensures that no interference occurs due to hidden terminals. Radios are half-duplex and can only tune on to one frequency at a time. There are N nodes in the system and M data channels (frequency hops) available, where $M > N$. This is the case for a typical multi-hop packet radio network operating on the ISM bands and using FHSS radios as described in Section 1, where the number of neighbors of each node is usually smaller than the available frequencies.

The channels are assumed to be error free and have no capture effect, so that collision of packets is the only source of errors, and more than one packet overlapped on the same channel at a receiver leads to a collision and no packets involved in it can be received correctly by the receiver.

It is assumed that data packets arrive at each node according to Poison process with average arrival rate λ. Each node has exactly one buffer for a data packet. The destination of any data packet from each node is assumed to be uniformly distributed among all its neighbors. All the nodes are synchronized and all channels are slotted with the slot size equal to η. Therefore, the total traffic load normalized to slot size is denoted by

$$G = N\lambda\eta$$

To simplify our comparative analysis, we ignore any propagation delay, guard time or any processing time. These parameters can be easily taken into account if necessary, and their values are far smaller than packet lengths in networks operating in ISM bands. Because IP packets have variable sizes, we are only interested in variable-length data packets. For tractability, we assume that any data packet is transmitted at the beginning of a HRMA slot and can only end at the end of a slot; therefore, the size of the data packet δ is a multiple of the slot size. We further assume that δ follows a geometric distribution with an average size of d slots, which implies that the probability that a data packet ends at the end of a slot is $q = 1/d$. We also denote the probability that a data packet does not end at a slot by $p = 1 - q$.

Throughput is defined as the the average utilization of the receiver (or transmitter) per node, i.e., the average percentage of time that each node receives (or transmits) data packets successfully. Because we assume half-duplex radios, the maximum throughput per node of any MAC protocol is 0.5.

3.2. HRMA Throughput

The length of HR, RTS or CTS is the same and is denoted by γ, and the size of the synchronization period is a multiple of γ, $(c-1)\gamma$. Thus, the slot size $\eta = (c+2)\gamma$. For simplicity, given that all protocols being considered rely on time slotting and would benefit from the same synchronization solution proposed for HRMA, we ignore the synchronization slot in our comparative analysis and assume that the synchronization period of a slot is much longer than the sum of RTS, CTS and HR period.

We use a Markov process to describe the operation of HRMA, where each state of the process represents the number of channels being used to transmit data in a slot. The maximum state is therefore $K = \lfloor N/2 \rfloor$. The state transition diagram for the Markov chain is show in Figure 4. Any state k, $0 \le k < K$, of the Markov chain can transit to any state $i \le k+1$. State K can transit to all states.

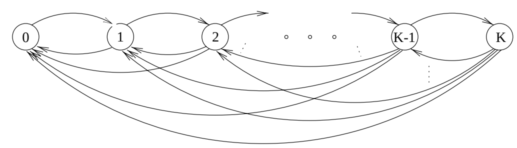

Figure 4: Markov process for HRMA

The probability that any node has data to send in the access period of a slot (i.e., the synchronization or HR period) is given by

$$p_a = 1 - e^{-c\gamma\lambda} \tag{1}$$

Let $A_k^{(i)}$ be the probability that i nodes attempt to transmit an RTS in state k, we have

$$A_k^{(i)} = \binom{N-2k}{i} p_a^i (1-p_a)^{N-2k-i} \tag{2}$$

Let $i = 1$, the probability that only one RTS arrives during the access period of a slot equals

$$A_k^{(1)} = (N-2k)(1-e^{-\frac{c}{c+2}\frac{G}{N}})(e^{-\frac{c}{c+2}\frac{G}{N}})^{N-2k-1} \tag{3}$$

The success probability of an RTS at any given state k, $P_{S|k}$, is equal to the probability that: (a) only one RTS arrives during the access period of a slot, (b) the current channel is not reserved for the future slot(s), and (c) the intended receiver is not transmitting or receiving a data packet. Therefore,

$$P_{S|k} = A_k^{(1)}(1-P_{R|k})(N-2k-1)/(N-1) \tag{4}$$

where $P_{R|k}$ is the probability that the current channel is reserved for the future slot(s) given that the system is in state k. It can be seen that this probability is less than the probability that any other channel is reserved for the future, given that the system is in state k, because the data packet needs to last longer on the current channel than on any other channel to reserve the channel. Thus we can have an upper bound of $P_{R|k}$ by letting

$$P_{R|k} = \frac{k}{M}p$$

which leads to a lower bound of the throughput of HRMA. This lower bound is very close to the actual value of the throughput when the number of the available channels are large compared to the total number of nodes, which can be seen in the numerical results presented in Section 4.

HRMA guarantees that any successful RTS leads to a successful data transmission and the probability of a data packet completing during a slot is q. At any given state $1 \le i \le K$, the probability that the next state is smaller than i is given by

$$\begin{aligned}\alpha_i &= P_{S|i}\sum_{j=2}^{i}\binom{i}{j}q^j p^{i-j} + (1-P_{S|i})\sum_{j=1}^{i}\binom{i}{j}q^j p^{i-j} \\ &= 1-p^i - iP_{S|i}qp^{i-1} \qquad (5)\end{aligned}$$

Because the Markov process is balanced across a cut between any state k and $k+1$, we obtain

$$\pi_k P_{S|k}p^k = \sum_{i=k+1}^{K}\pi_i\alpha_i \qquad (6)$$

where π_k is the probability of state k and $0 \le k \le K-1$. If we let

$$Q_i = \frac{\pi_i}{\pi_K}$$

(thus $Q_K = 1$) after arrangement Equation (6) becomes

$$Q_k = \frac{\sum_{i=k+1}^{K} Q_i\alpha_i}{P_{S|k}p^k} \quad 0 \le k \le K-1 \qquad (7)$$

Because

$$\sum_{i=0}^{K}\pi_i = 1$$

it yields

$$\pi_K = \frac{1}{1+\sum_{i=0}^{K-1}Q_i} \qquad (8)$$

and we can get the state probabilities by

$$\pi_k = \pi_K Q_k \qquad (9)$$

Finally, the throughput of HRMA is

$$S = \frac{\sum_{k=1}^{K}\pi_k k}{N} \qquad (10)$$

3.3. Throughput of Ideal Multichannel Slotted Access Protocol

For comparison purposes, we consider the following ideal multichannel slotted channel access protocol. Each node is assigned a unique channel (frequency hop) to which it is tuned when it is not transmitting, and the node tunes its radio to the channel of the intended receiver to transmit a packet, which is usually called receiver-oriented channel assignment (ROCA). When a node has a packet to send, it attempts to transmit in the next slot. There exist two possible types of conflict. One is that two or more nodes try to start sending packets to the same receiver at the same slot. The other one is that the destination is transmitting or receiving. We assume that when the first conflict happens, the ideal protocol can randomly pick one competing sender to occupy the destination's channel and transmit if the destination is not among these attempting senders and is ready to receive in the next slot; and it will block all the attempting senders when the second case happens. Therefore, there is no collision in the channels. The only issue that affects the throughput is the pair-up of nodes.

We can use the Markov chain shown in Figure 5 to describe the operation of the ideal protocol, where each state of the chain represents the number of channels being used to transmit data packets during a slot. Let π_k denote the probability of state k, $0 \le k \le K, K = \lfloor N/2 \rfloor$. According to our assumptions, each state of the Markov chain can transit to any state. A transition may occur in the next slot when nodes finish transmitting or idle nodes have arrivals.

Before proceeding further we introduce the following notations. Assume that in a given slot t the system is in state k. The

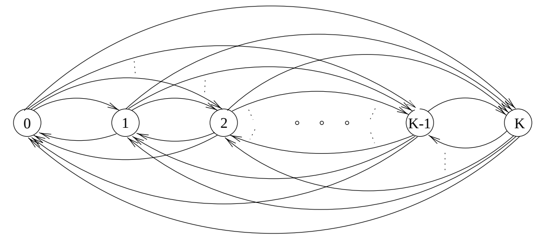

Figure 5: Markov process for ideal protocol and ALOHA

probability that n nodes $(0 \le n \le k)$ finish transmitting at the end of slot t is denoted by $D_k^{(n)}$. The probability that m nodes $(0 \le m \le N-2k)$ have packet arrivals and attempt to transmit in the next slot $t+1$ is denoted by $A_k^{(m)}$. According to this notation, we have

$$D_k^{(n)} = \binom{k}{n}q^n(1-q)^{k-n} \quad 0 \le n \le k \qquad (11)$$

and $A_k^{(m)}$ can be expressed by Equation (2) except that p_a is given by

$$p_a = 1-e^{-G/N} \qquad (12)$$

instead of Equation (1). In slot t, all the nodes in the system can be partitioned into four sets: S_b, which consists of the nodes that are transmitting or receiving data and will not finish in slot t, and has $2k-2n$ members; S_d, which contains the nodes that are transmitting or receiving data but will finish in slot t, and has $2n$ members; S_a, which consists of the nodes that are idle and have packet arrivals in slot t, and has m members; and S_i, which contains the nodes that are idle and have no packet arrivals in slot t, and has $N-2k-m$ members. When calculating the transition probabilities, we will condition on the number of members in S_d. For the transition from state k in slot t to state l in slot $t+1$, at least $\hat{n} = max(0, k-l)$ nodes must finish sending in slot t; therefore, $n \ge \hat{n}$, and $\hat{m} = l-(k-n)$ nodes should become successful senders in slot $t+1$ and $m \ge \hat{m}$. It can be seen that, in slot $t+1$, all the members in S_b or S_a will not be available for receiving new packets while all the members in S_i or S_d will be available for receiving new packets. Therefore, the total number of nodes that will not be available for receiving new packets in slot $t+1$ is $N_{na}(k,n,m) = 2k-2n+m$ and the total number of nodes that will be available for receiving new packets in slot $t+1$ is $N_{av}(k,n,m) = N-2k+2n-m$.

Denote by $S_{k,n,m}(\hat{m})$ the probability that $\hat{m}$ new arrivals in slot t will be successful in slot $t+1$ given that, in slot t, k nodes are sending data and n of them will finish and m idle nodes have arrivals. We calculate $S_{k,n,m}(\hat{m})$ using a mapping concept. To have $\hat{m}$ new successful senders in slot $t+1$, it should hold true that $N_{av}(k,n,m) \ge \hat{m}$, and any node in S_a must be mapped (addressed) to other nodes in such a way that exactly $\hat{m}$ members in $S_i \cup S_d$ are mapped. Denote by $C_{k,n,m}(\hat{m})$ the total number of such desired mappings.

For $N_{av}(k,n,m) < \hat{m}$, $S_{k,n,m}(\hat{m}) = 0$. Therefore, we only need to consider the cases where $N_{av}(k,n,m) \ge \hat{m}$. If $N_{na}(k,n,m)$ equals 0, m and $\hat{m}$ must both be 0, which means that $S_{k,n,m}(\hat{m}) = 1$. If $N_{na}(k,n,m)$ equals 1, m must be 1; therefore, all neighbors are available and thus $S_{k,n,m}(\hat{m}) = 0$ if $\hat{m} = 0$ and $S_{k,n,m}(\hat{m}) = 1$ if $\hat{m} = 1$. In the following, we consider the cases where $N_{av}(k,n,m) \ge \hat{m}$ and $N_{na}(k,n,m) > 1$. It is also immediate that

$$C_{k,n,m}(0) = [N_{na}(k,n,m)-1]^m = \Theta_{k,n,m}(0) \qquad (13)$$

In general, for any $0 \le i \le \hat{m}$, we can obtain $\Theta_{k,n,m}(i)$, the total number of mappings where exactly i given members from

$S_i \cup S_d$ are mapped, is

$$\Theta_{k,n,m}(i) = [N_{na}(k,n,m) + i - 1]^m - \sum_{j=0}^{i-1} \binom{i}{j} \Theta_{k,n,m}(j) \tag{14}$$

It follows that

$$C_{k,n,m}(i) = \binom{N_{av}(k,n,m)}{i} \Theta_{k,n,m}(i) \tag{15}$$

There are a total of $(N-1)^m$ possible mappings if there are m arrivals. Therefore,

$$S_{k,n,m}(\hat{m}) = \frac{C_{k,n,m}(\hat{m})}{(N-1)^m} \tag{16}$$

The transition probability from state k to l is then given by

$$P_{lk} = \sum_{n=\hat{n}}^{k} D_k^{(n)} \sum_{m=\hat{m}}^{N-2k} A_k^{(m)} S_{k,n,m}(\hat{m}) \tag{17}$$

We can solve the global balance equations with

$$\pi_l = \sum_{k=0}^{K} \pi_k P_{lk}$$

and the condition

$$\sum_{l=0}^{K} \pi_l = 1$$

which yields the throughput of the system using Equation (10).

3.4. Throughput of Multichannel Slotted ALOHA

Prior MAC protocols based on slow FHSS assume ALOHA or slotted ALOHA access to the channel and typically assume ROCA (e.g., Metricom's system [3]). We consider here a slotted ALOHA with ROCA. To keep tractable, we further assume that transmitting has the highest priority and preempts any receiving. When a packet arrives at a node not transmitting, it will be transmitted at the beginning of the next slot.

For any given node (on a specific frequency) we can construct a queue system with $N-1$ customers and $N-1$ servers. The arrival probability at each node in any slot is the same as Equation (12). The service time for each arrival is the packet length. We can use the same Markov model (see Figure 5) as that used in the ideal protocol to obtain the state probabilities $\pi_k, 0 \le k \le N-1$, where the state is the number of busy servers in a slot. However, here we have N states and the state transition probability from state k to state l is given by:

$$P_{lk} = \sum_{n=\hat{n}}^{k} A_k^{(n+l-k)} D_k^{(n)} \tag{18}$$

where

$$A_k^{(m)} = \binom{N-k-1}{m} p_a^m (1-p_a)^{N-k-m-1}$$

$D_k^{(n)}$ is given by Equation (11) and p_a is given by Equation (12).

Denote by $B_i^{(j)}$ the probability that j nodes are sending packets to node R given that i nodes are transmitting, which can been expressed by

$$B_i^{(j)} = \binom{i}{j} \left(\frac{1}{N-1}\right)^j \left(\frac{N-2}{N-1}\right)^{i-j} \tag{19}$$

The successful receiving probability for any node R is equal to the probability that: (a) only one packet is directed to node R from its neighbors in a slot, (b) all the current packet(s) being transmitted to or from node R if any end during this slot, and (c) no other packet(s) will be transmitted to or from node R during its receiving time. To keep the analysis tractable, we assume that during the receiving time of any packet at R, any node can send at most one packet, which leads to an upper bound of the throughput of ALOHA with ROCA, because we underestimate the collision probability. The probability that an idle neighbor of node R has no packet arrival for R in i consecutive slots, denoted by E_i, is

$$E_i = 1 - \frac{1 - e^{-iG/N}}{N-1} \tag{20}$$

It follows that the probability of (c), when R has r idle neighbors, is

$$C_r = \sum_{s=1}^{\infty} p^{s-1} q (1-p_a)^{s-1} E_{s-1}^r \tag{21}$$

The throughput for any node that is not transmitting when the packet arrives is

$$S_1 = \sum_{k=0}^{N-2} \pi_k \sum_{m=1}^{N-k-1} A_k^{(m)} B_m^{(1)} \sum_{j=0}^{k} B_k^{(j)} q^j \sum_{n=0}^{k-j} D_{k-j}^{(n)} C_x \tag{22}$$

where $x = N-k-m+j+n-1$.

Any node in a slot must be either transmitting or not transmitting, and we can use a simple two-state Markov chain with p_a as the transition probability from the non-transmitting state to the transmitting state and q as the transition probability for the reverse direction to describe its behavior. Solving this Markov chain, we get the transmitting probability for any node in a slot to be

$$P_t = \frac{p_a}{p_a + q}$$

Therefore, the throughput is

$$S = (1-P_t)S_1 + P_t q S_1 \tag{23}$$

4. Numerical Results

The numerical results are given in Figure 6 through Figure 11, which depict the throughput per node (S) as a function of offered load (G) with different numbers of nodes (N), different values of average packet length (APL) and different numbers of channels available (for the case of HRMA) to reflect the effect of different choices of the network parameters on the performance.

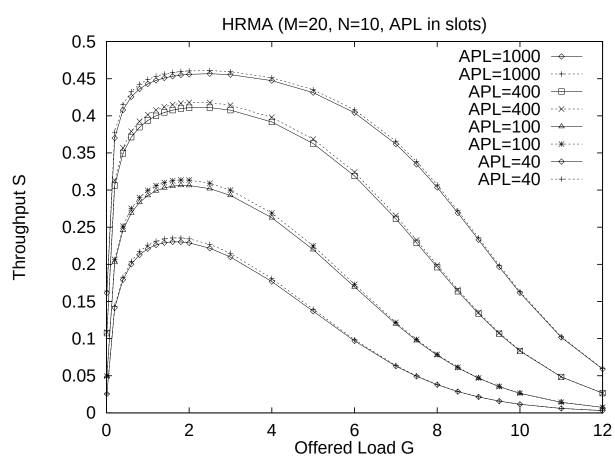

Figure 6: Throughput of HRMA with different values of APL

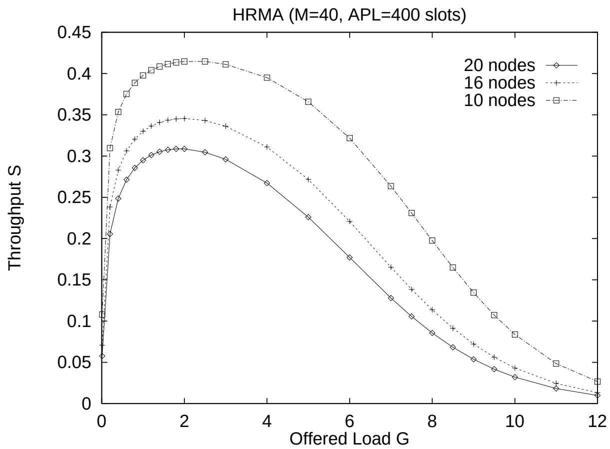

Figure 7: Throughput of HRMA with different numbers of nodes

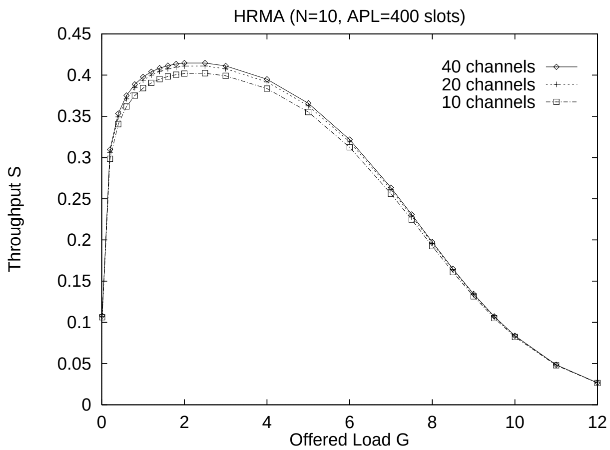

Figure 8: Throughput of HRMA with different numbers of channels

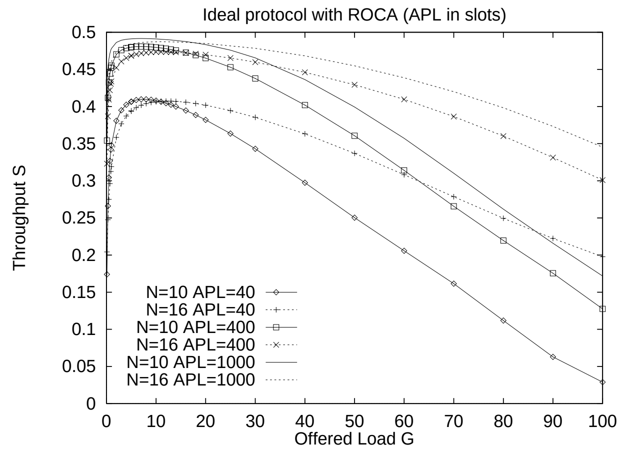

Figure 9: Throughput of Ideal protocol with different population and APL's

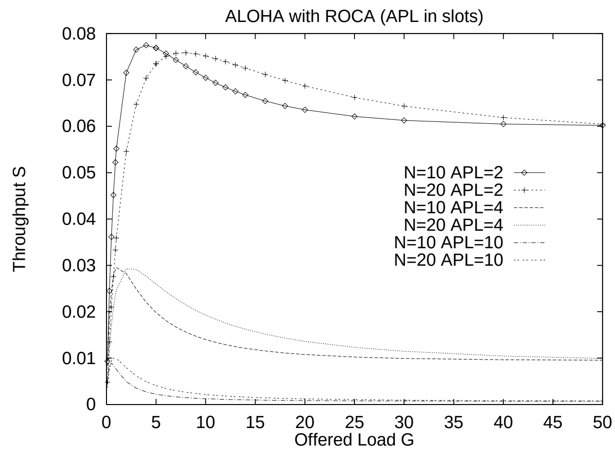

Figure 10: Throughput of ALOHA with different population and APL's

Figure 6 plots the throughput of HRMA with different values of average packet length in slots, where the system has 10 nodes and 20 available channels. Throughput grows significantly when the APL increases, with the maximum throughput being close to the theoretical maximum value. This is because HRMA eliminates data collisions; once successful, a large data packet can reserve the channel for a long time, which greatly reduces the overhead and improves the utilization of the channel. HRMA is more attractive with large packets or packet trains. The curves plotted with dashed lines show the upper bounds of the throughput with the assumption that the channel is always available for the RTS-CTS handshake. It can be seen that the upper bounds are very close to our approximation.

The throughput of HRMA with average packet length of 400 slots and 40 available channels is displayed in Figure 7 for the case of systems with 10, 16, and 20 nodes. The curves indicate that the throughput increases as the number of nodes decreases, which is expected. Because HRMA uses a common signaling channel, having more nodes in the systems leads to more collision on the signaling channel, which reduces the throughput.

In Figure 8, we show the effect of changing the number of available channels on the throughput. The system has 10 nodes and the APL equals 400 slots. Adding channels adds little performance improvement. HRMA allows idle nodes to contend for every unreserved hop by sending RTS's on the unreserved hop. As long as the number of available channels exceeds the number of nodes, the success probability for RTS's will not change much with additional channels. Again, we see that the APL plays a very important role on performance. Systems in our examples with the same APL almost show the same throughput, even with the different number of channels available.

Figure 9 and Figure 10 show the throughput for the ideal protocol and ALOHA, respectively, each with different values of APL and different numbers of nodes. In this paper we assume that the systems have finite population and multiple channels, where pair-up is also an important issue that can affect the throughput. When the offered load is high, few nodes are available for receiving; therefore, the throughput decreases as the traffic load increases, even for the ideal protocol. Due to the same reason, the performance of the ideal protocol is better with a larger APL. In contrast, a larger APL leads to more collisions and thus lower throughput for the case of ALOHA. The throughput for ALOHA is very low even if the APL is small.

Figure 11 compares the throughput performance of three protocols for the systems with 10 nodes and 20 channels. The graphs show that in the traffic-load range of interest and with large average packet length compared to the slot size, HRMA performs much better than ALOHA. Moreover, HRMA has the potential to get close to the performance of the ideal protocol with very large packet sizes or packet trains.

5. Conclusions

We have described a new multichannel MAC protocol for adhoc networks (multihop packet-radio networks) and analyzed its performance. HRMA dynamically allocates frequency bands to nodes using a common frequency-hopping pattern, such that data

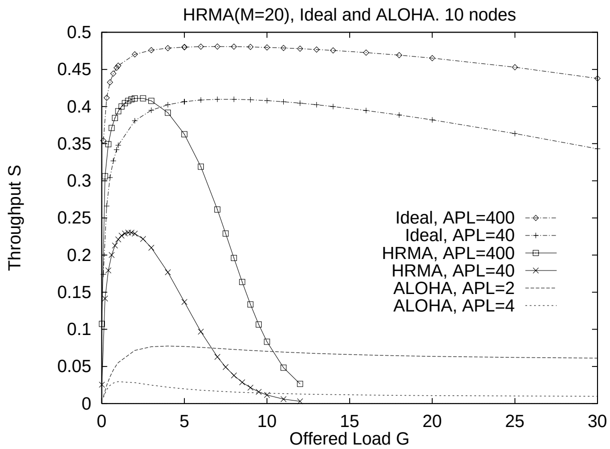

Figure 11: Throughput comparison: HRMA, Ideal and ALOHA

and acknowledgements are transmitted without hidden-terminal interference. HRMA allows systems to merge and nodes to join existing systems. HRMA's features are achieved using simple half-duplex FHSS radios commercially available today. Our analysis shows that HRMA's throughput performance is significantly better than slotted ALOHA with ROCA, which is representative of the current practice using commercial radios. HRMA can achieve a maximum throughput that is comparable to that of the ideal protocol in which a receiver is always able to receive one transmission from multiple senders, especially when data packets are large compared to the slot size used for frequency hopping. This high throughput is obtained without the need for complex code assignment through a very simple reservation mechanism.

Our work continues to analyze the performance of HRMA in multi-hop packet-radio networks and to develop and analyze variants of HRMA with improved performance.

6. References

[1] *P802.11–Unapproved Draft: Wireless LAN Medium Access Control (MAC) and Physical Specifications*. IEEE, January 1996.

[2] *FCC Rules and Regulations, Part15.247 and 15.249*. October 1997.

[3] *Metricom*. http://www.metricom.com, Los Gatos, CA, 1998.

[4] *WINGS for the Internet Project and SPARROW Project*. http://www.cse.ucsc.edu/research/ccrg/, Santa Cruz, CA, 1998.

[5] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. Macaw: A media access protocol for wireless lan's. In *Proceedings ACM SIGCOMM*, pages 212–25, London, UK, August 1994.

[6] A. Ephremides, J. E. Wieselthier, and D. J. Baker. A design concept for reliable mobile radio networks with frequency hopping signaling. *Proceedings of the IEEE*, 75(1):56–73, January 1987.

[7] C. L. Fullmer and J. J. Garcia-Luna-Aceves. Solutions to hidden terminal problems in wireless networks. In *Proceedings ACM SIGCOMM*, Cannes, France, September 1997.

[8] M. B. Pursley. The role of spread spectrum in packet radio networks. *Proceedings of the IEEE*, 75(1):116–34, January 1987.